eContracting

Andreas Meier & Luis Terán
Information Systems Research Group
University of Fribourg
Learning Targets

- Elements of eContracting
- Generic Services for the negotiation process
- Authentication and user management
- Symmetric encryption
- DES
- Asymmetric encryption
- Cryptographic hash functions
- SHA
- Message Authentication Code (MAC)
- Digital signatures
- Public Key Infrastructure (PKI)
• **Contract**: legally binding agreement between two or more people on a statement of intention.
• Electronics Agreements requires unmistakable proof of origin for electronic contracts (electronics signatures).
• Electronic Contract: legally binding digital documents
  - Who are the contract partners?
  - What is the content of the agreement?
  - How are the electronically determined obligations put into practice?
  - What are the determined goods or services due?
  - What legal framework conditions apply?
Electronic Negotiation Process

- In eContracting the electronic negotiation involves the following actions:
  - Valid recording of the negotiation positions
  - Electronic storage and administration of the contract parts
  - Arrangement on rights and obligations
  - Legally binding conclusion of a contract (with digital signatures)
  - Monitoring of the fulfilment of terms of the contract
Generic services for the Negotiation Process

Electronic representation of providers
Electronic representation of customers
Signature service
Negotiation service
Validation service
Enforcement service
Arbitration service
Electronic payment service
Electronic logistics service

Generic services for eContracting by Runge (2000)

Time
Generic services for the Negotiation process

- Identification of parties through CA. To issue certificate, CA requires identifications (passport, contacts, etc)

  - Electronic representation of providers
  - Electronic representation of customers

  - Electronic payment service
  - Electronic logistics service

- Signature service
- Enforcement service
- Validation service
- Arbitration service

- Information phase
- Agreement phase
- Execution phase

Generic services for eContracting by Runge (2000)
Generic services for the Negotiation Process

Checks the contract framework
Highlight problems and risks
Suggest Notifications

Signature service

Enforcement service

Electronic representation of providers

Electronic representation of customers

Negotiation service

Validation service

Archiving service

Arbitration service

Electronic logistics service

Electronic payment service

Information phase

Agreement phase

Execution phase

Generic services for eContracting by Runge (2000)
Generic services for the Negotiation Process

Supports negotiation through multimedia components and cooperative negotiation environments

- Signature service
- Enforcement service
- Arbitration service
- Electronic payment service
- Electronic logistics service

Electronic representation of providers
Electronic representation of customers
Archiving service
Validation service

Information phase → Agreement phase → Execution phase

Generic services for eContracting by Runge (2000)
Use electronic catalogs to optimize conditions for exchange
Classification of contract versions

Electronic representation of providers
Electronic representation of customers

Electronic payment service
Electronic logistics service

Generic services for eContracting by Runge (2000)

Information phase
Agreement phase
Execution phase
Generic services for the Negotiation Process

- Electronic representation of providers
- Electronic representation of customers
- Negotiation service
- Validation service
- Archiving service
- Signature service
- Enforcement service
- Arbitration service
- Electronic logistics service
- Electronic payment service

To solve legal disputes (online arbitration court)
http://www.cybersettle.com

Information phase → Agreement phase → Execution phase

Generic services for eContracting by Runge (2000)
Generic services for the Negotiation Process

Generic services for eContracting by Runge (2000)
Identity Management supports the following task:
- Identification → determine a unique identification of user/contractor
- Authentication → verification of the authenticity of the user/contractor (digital signatures)
- Authorization → processing capabilities (access rights and/or change of law).

Identifications is not only important in eContracting. In the eGovernment Framework is also used in:
- eProcurement
- ePayment
- eCommunity
- eDemocracy (eVoting and eElections)
Role-based Access Control (RBAC)

- Developed at the National Institute of Standards and Technology (US)
- Components:
  - Users (can have different roles)
  - Roles (privileges are defined by a user to perform his duties, a user can have different roles)
  - Operations
  - Objects
  - Permissions
Role-based Access Control (RBAC)

Possible values in the RBAC-model
The SZ42 Lorenz cipher machine, used in World War II to encrypt very-high-level general staff messages [Wikipedia]
**Cryptography** is the practice and study of hiding information
- *Encryption* is the process of converting ordinary text (*plaintext*) into unintelligible gibberish (*ciphertext*).
- *Decryption* is the process of converting the unintelligible *ciphertext* back to *plaintext*.

**Types of cryptographic algorithms:**
- *Symmetric*: uses a single key for both, encryption and decryption.
- *Asymmetric*: uses two keys, one for encryption (public key) and other for decryption (private key).
- *Hash functions*: deterministic that converts an arbitrary block into a fixed-size bit string, the (cryptographic) *hash value*.
- *Message Authentication*: guaranties authenticity of messages
Symmetric Cryptography

• Single Key is used for encryption and decryption
• Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers RC4, GSM–A5/1, Bluetooth–E0, CSS, etc
• Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4 GSM–A5/1 Bluetooth–E0 CSS, etc
- Problems:
  - Key must be shared in a *secure way* (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

• Single Key is used for encryption and decryption
• Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4 GSM–A5/1 Bluetooth–E0 CSS, etc
• Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: **Block Ciphers** (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and **Stream Ciphers** RC4, GSM–A5/1, Bluetooth–E0, CSS, etc
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: **Block Ciphers** (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and **Stream Ciphers** (RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication

### Secure channel

Bob

Alice
Symmetric Cryptography

• Single Key is used for encryption and decryption
• Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers (RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
• Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication

Secure channel

Bob

Alice
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* (RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4 GSM–A5/1 Bluetooth–E0 CSS, etc)
- Problems:
  - Key must be shared in a *secure way* (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: **Block Ciphers** (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and **Stream Ciphers** RC4, GSM–A5/1, Bluetooth–E0, CSS, etc
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
**Symmetric Cryptography**

- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4, GSM–A5/1, Bluetooth–E0, CSS, etc
- Problems:
  - Key must be shared in a **secure way** (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4 GSM–A5/1 Bluetooth–E0 CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers RC4, GSM–A5/1, Bluetooth–E0, CSS, etc
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication

Bob

ENC

Alice

DEC
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
- Single Key is used for encryption and decryption
- Categories: *Block Ciphers* (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and *Stream Ciphers* RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
- Problems:
  - Key must be shared in a *secure way* (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers RC4, GSM–A5/1, Bluetooth–E0, CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: **Block Ciphers** (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and **Stream Ciphers** RC4 GSM–A5/1 Bluetooth–E0 CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Symmetric Cryptography

- Single Key is used for encryption and decryption
- Categories: Block Ciphers (DES, 3DES, IDEA, BLOWFISH, RC5, AES, etc) and Stream Ciphers RC4 GSM–A5/1 Bluetooth–E0 CSS, etc)
- Problems:
  - Key must be shared in a secure way (how to do that?)
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

• Set a pair of keys: **public key** (available for everybody), **private key** (secret)
• Some algorithms: · DH · DSA · ElGamal (encryption · signature scheme) · RSA · SRP · STS · Three-pass protocol · XTR
• Problems:
  - Integrity and Authentication

Internet

Bob

Alice
Public Key (Asymmetric) Cryptography

• Set a pair of keys: public key (available for everybody), private key (secret)
• Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
• Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: DH, DSA, ElGamal (encryption · signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: · DH · DSA · ElGamal (encryption · signature scheme) · RSA · SRP · STS · Three-pass protocol · XTR
- Problems:
  - Integrity and Authentication
**Public Key (Asymmetric) Cryptography**

- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: · DH · DSA · ElGamal (encryption · signature scheme) · RSA · SRP · STS · Three-pass protocol · XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication

Bob

Alice
Public Key (Asymmetric) Cryptography

- Set a pair of keys: public key (available for everybody), private key (secret)
- Some algorithms: DH, DSA, ElGamal (encryption, signature scheme), RSA, SRP, STS, Three-pass protocol, XTR
- Problems:
  - Integrity and Authentication
PUBLIC KEY (ASYMMETRIC) CRYPTOGRAPHY

- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: · DH · DSA · ElGamal (encryption · signature scheme) · RSA · SRP · STS · Three-pass protocol · XTR
- Problems:
  - Integrity and Authentication
PUBLIC KEY (ASYMMETRIC) CRYPTOGRAPHY

• Set a pair of keys: public key (available for everybody), private key (secret)
• Some algorithms: DH · DSA · ElGamal (encryption · signature scheme) · RSA · SRP · STS · Three-pass protocol · XTR
• Problems:
  - Integrity and Authentication
Cryptographic Hash Functions

- Deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the Hash Value. HF are used for Message Authentication Codes, Digital Signature, etc.

- Properties:
  - Preimage Resistance
    - Given $h$ it should be hard to find a message $m$ such that: $h = \text{hash}(m)$
  - Second Preimage Resistance
    - Given $m_1$ it should be hard to find $m_2$ where $m_1 \neq m_2$ such that: $\text{hash}(m_1) = \text{hash}(m_2)$
  - Collision Resistance
    - It should be hard to find two different $m_1$ and $m_2$ such that: $\text{hash}(m_1) = \text{hash}(m_2)$

- Some algorithms: MD5 | SHA-1 | SHA-2 | SHA-3
Cryptographic Hash Functions

• Deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the Hash Value. HF are used for Message Authentication Codes, Digital Signature, etc.

• Properties:
  - Preimage Resistance
    ‣ Given \( h \) it should be hard to find a message \( m \) such that: \( h = \text{hash} (m) \)
  - Second Preimage Resistance
    ‣ Given \( m_1 \) it should be hard to find \( m_2 \) where \( m_1 \neq m_2 \) such that: \( \text{hash}(m_1) = \text{hash} (m_2) \)
  - Collision Resistance
    ‣ It should be hard to find two different \( m_1 \) and \( m_2 \) such that: \( \text{hash}(m_1) = \text{hash} (m_2) \)

• Some algorithms: MD5 | SHA-1 | SHA-2 | SHA-3

Arbitrary size

<table>
<thead>
<tr>
<th>Input</th>
<th>Hash Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>146B 75CA B124 5681 CD12 4834 A34C BA67</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog.</td>
<td>AE2D 59A0 BBB4 9801 639B 5648 00A1 1945</td>
</tr>
<tr>
<td>The redd fox jumps over the blue dog.</td>
<td>FF56 C349 80CA 345A EE78 31AB 98CE 2210</td>
</tr>
</tbody>
</table>

Fixed-size
Cryptographic Hash Functions

- Deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the Hash Value. HF are used for Message Authentication Codes, Digital Signature, etc.

- Properties:
  - Preimage Resistance
    ‣ Given \( h \) it should be hard to find a message \( m \) such that: \( h = \text{hash} (m) \)
  - Second Preimage Resistance
    ‣ Given \( m_1 \) it should be hard to find \( m_2 \) where \( m_1 \neq m_2 \) such that: \( \text{hash}(m_1) = \text{hash} (m_2) \)
  - Collision Resistance
    ‣ It should be hard to find two different \( m_1 \) and \( m_2 \) such that: \( \text{hash}(m_1) = \text{hash} (m_2) \)

- Some algorithms: MD5 \| SHA-1 \| SHA-2 \| SHA-3

---

**Input**

- \text{Hello}
- The red fox jumps over the blue dog.
- The redd fox jumps over the blue dog.

**Hash Value**

- \( 146B \ 75CA \ B124 \ 5681 \ CD12 \ 4834 \ A34C \ BA67 \)
- \( AE2D \ 59A0 \ BBB4 \ 9801 \ 639B \ 5648 \ 00A1 \ 1945 \)
- \( FF56 \ C349 \ 80CA \ 345A \ EE78 \ 31AB \ 98CE \ 2210 \)

**Arbitrary size**

**Fixed-size**
This standard specifies four secure hash algorithms, SHA-1, SHA-256, SHA-384, and SHA-512.
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)

Secure channel

Bob

Alice
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)

Secure channel

Bob

Alice
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)

Bob

Hello

MAC Algorithm

Alice

MAC Algorithm
**Message Authentication Codes (MACs)**

- Used to **authenticate** messages. It is also called **Keyed Hash Function**.
- Some algorithms: DAA I CBC-MAC I HMAC I OMAC/CMAC I PMAC.
- Problems:
  - Key must be shared in a **secure way** (how to do that?)

Hello

Bob

MAC Algorithm

Alice

MAC Algorithm
Message Authentication Codes (MACs)

• Used to authenticate messages. It is also called Keyed Hash Function.
• Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
• Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

• Used to authenticate messages. It is also called Keyed Hash Function.
• Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
• Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

• Used to authenticate messages. It is also called Keyed Hash Function.

• Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.

• Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

- Used to **authenticate** messages. It is also called **Keyed Hash Function**.
- **Some algorithms**: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- **Problems**:
  - Key must be shared in a **secure way** (how to do that?)
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)
Message Authentication Codes (MACs)

- Used to authenticate messages. It is also called Keyed Hash Function.
- Some algorithms: DAA | CBC-MAC | HMAC | OMAC/CMAC | PMAC.
- Problems:
  - Key must be shared in a secure way (how to do that?)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)

Internet (CA)

Bob

Alice
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
**Digital Signatures**

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

• Used to demonstrate the **authenticity** of a document
• Set a pair of keys: public key (available for everybody), private key (secret)
• Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
**Digital Signatures**

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
• Used to demonstrate the **authenticity** of a document
• Set a pair of keys: **public key** (available for everybody), **private key** (secret)
• Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
**Digital Signatures**

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
**Digital Signatures**

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- **Some algorithms:** RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Digital Signatures

- Used to demonstrate the **authenticity** of a document
- Set a pair of keys: **public key** (available for everybody), **private key** (secret)
- Some algorithms: RSA · DH · DSA · EKE · ElGamal (encryption · signature scheme)
Public Key Infrastructure

- A asymmetric encryption system for communication
- A framework not a specific technology
- Can work across multiple systems and vendors
- Provides authentication and confidentiality
  - Authentication: confirms the owner of the keys using DIGITAL CERTIFICATES
  - Confidentiality: encrypts data transmission
Public Key Infrastructure

• Digital Certificates
  - Helps with authentication.
  - Associates a public key with an individual/company.
  - Issued by a **Certificate Authority (CA)**.
  - Most common standards:
    ‣ Protocols: SSL, IPSec, S/MIME, Privacy Enhanced Mail (PEM), or SET.
    ‣ Certificate format: X.509, PGP
• Certificate Authority (CA)
  - Collection of personnel and computer systems
    ‣ Highly secured (firewalls, etc)
    ‣ Strong management controls (separation of duties)
  - Responsible for issuing, revoking, and distributing certificates
  - Often a trusted third-party organization. Commercial examples:
    ‣ DigiCert (http://www.digicert.com)
    ‣ VeriSign (http://www.verisign.com)
  - Companies or organizations can have an in-house CA
  - Stores the public key in a directory available to anyone
• **Registration Authority (RA)**
  - Perform the necessary checks on the person or company requesting the certificate.
  - Proof of ID
  - Identification of applicant
  - Approval of Pseudonyms

• **Validation Authority (VA)**
  - Can provide information on behalf of CA (third-party)
Public Key Infrastructure

1. RA
2. CA
3. User
4. VA
5. Secure Certificate
6. eShop